AI-assisted Software Tool to Visualize Flood Forecasting with Water Routing

Design Document

Team Number 37

Client: Aishwarya Sarkar - 3rd Year Ph.D

Client / Faculty Advisor: Dr. Ali Jannesari

Team Members: Ani Manjunath, Gabriel Rau, Quinn Conrad, Siyu Wang, Kylus Pettit-Ehler, Eric Korneisel, Ryan Thompson

Team Email: sdmay22-37@iastate.edu

Team website: http://sdmay22-37.sd.ece.iastate.edu/

Revised: 12/5/21

Executive Summary

Development Standards & Practices Used

- Using an agile development process during implementation (based around client)
- Testing will be verified manually (hard to automate)
- Version control will handled by using GIT
- Data that goes through our model securely handled
- Users should be viewing information that is coherent and reliable
- Emphasis on a minimum-cost, effective solution
- Software developed should be sustainable (usable by others)
- All frameworks used given due credit

Summary of Requirements

- Selectable Watersheds
- Interactive Map
- Mouse-over coordinates, drag and click map, real-time maps data
- Visual graphing of data
- Precipitation, temperature, land cover, soil properties, etc...
- Display of water discharge over time
- Color coordinated weather data and timeline
- Handle advanced datasets and new data
- Manageable response time and refresh rate
- Implementation of other map data like weather

Applicable Courses from Iowa State University Curriculum

- COM S 309 (software development)
- SE 329 (project management)
- COM S 363 (databases)
- SE 339 (software architecture)
- COM S 228 (data structures)
- SE 319 (Web development)

New Skills/Knowledge acquired that was not taught in courses

- Using a geographic information system (QGIS) to graph data
- Using a shapefile to map watershed data to specific regions
- Using a frontend framework (cakePHP) to query a database
- Using spatial and geographical objects in SQL (PostGIS)
- Load GeoJSON files with interactive layers using Leaflet
- Using front end frameworks to create an interface for a geographical map

Table of Contents

List of Figures / Footnotes

1 Team

1.1 Team Members

Ani Manjunath
Siyu Wang
Quinn Conrad
Kylus Pettit-Ehler
Ryan Thompson
Gabriel Rau
Eric Korneisel

1.2 Required Skill Sets for Your Project

Frontend - Creation of visual map and method of displaying the map using frontend languages

Backend - Migrating data to a database. Scripts to update it.

Version Control - Git

Testing - Unit tests

1.3 Skill Sets covered by the Team

Ani: Frontend/Full-stack - server management, UI, SQL, CI/CD, DevOps

Ryan: Backend - database, embedded systems

Quinn: Frontend/backend - UI/UX design, SQL

Kylus: Frontend/backend - UI development, backend databases, embedded systems

Gabe: Systems - Linux, automation, git

Siyu: Frontend/backend - UI, SQL

Eric: Frontend/Backend - UI design, SQL

1.4 Project Management Style Adopted by the team

Waterfall method for planning and then migrating to Agile during implementation.

1.5 Initial Project Management Roles

Ani: Client interaction - answering questions and coordinating with the client.

Ryan: Team organization - planning meetings.

Siyu: Deadline enforcement.

Quinn: In charge of meeting minutes/reports.

Gabe: Team communication leader.

Kylus: Conflict arbitrator.

Eric: Product Tester

2 Introduction

2.1 Problem Statement

Our client has been working on a machine learning model that attempts to predict watershed discharge at the end of river basins given inputs like precipitation, land cover, etc. Our goal is to help visualize this numerical data with an interactive map and customizable overlays.

2.2 Requirements & Constraints

Functional Requirements:
- Selectable Watersheds
- Interactive Map
- Mouse-over coordinates, drag and click map, real-time maps data
- Visual graphing of data
- Precipitation, temperature, land cover, soil properties, etc...
- Display of water discharge over time
- Color coordinated weather data and timeline

Non-Functional Requirements:
- Scalable
- Handle advanced datasets and new data
- Good performance
- Manageable response time and refresh rate
- Easy to maintain/update
- Reliable
- Implementation of other map data like weather

Constraints:
- Using format of existing datasets
- Access to datasets
- Licenses for external software to use as a project framework

2.3 Engineering Standards

- Any restrictions introduced by QGIS and Leaflet.js.. Right now we don't really see anything big in QGIS or Leaflet.js TOS. A few examples might be:
    - Our app must not be used for monetary gain as QGIS or Leaflet.js
    - Users must be notified if we collect location data from them
    - Our implementation must be easily identifiable by QGIS or Leaflt.js (we aren't hiding it from them)

- Any standards that may be present on flood hazard reporting (FEMA, etc). Most flood reporting requirements come from insurance agencies, so there may not be a lot of overlap on these requirements. Here are a couple examples:
    - Being sure data is well monitored and reliable
    - Making sure data is presented clearly and is not confusing
    - Data should be as “open” or as available as possible

2.4 Intended Users and Uses

The intended user for this project is our clients, a third year PHD student and Dr. Ali Jannesari. This project will be used to create an interactive UI for flood forecasting specifically for the data that the clients have gathered. The user should be able to have real-time Map data with grids and watersheds overlaid. They should get mouse-over coordinates and be able to drag around and navigate the map. The user can also view graph data gathered by a machine learning algorithm that is visualized within our UI.

3 Project Plan

3.1 Project Management/Tracking Procedures

We will be using waterfall planning with an agile development process. We have a defined plan for the project, but need to be flexible around the client.

Furthermore, we will use Gitlab and Trello to track tasks that need to be completed. Gitlab will provide a place to host our work, and Trello will more clearly outline which tasks are to be completed by which member.

3.2 Task Decomposition

- Frontend development
  - QGIS and Leaflet.js API integration
    - Different map overlays
  - UI features
    - Color coded timeline
    - Color coded map
    - Tabs/sections for graphs and overlays
  - Data visualization
    - Precipitation graphing
    - Graphing data in correct locations

- Backend development
  - Acquire data
    - Model with static data
    - Model with live data
  - Storing data
    - Connect server to database
  - Developing communication model with the frontend
    - Well-defined endpoints
    - Mapping frontend to backend

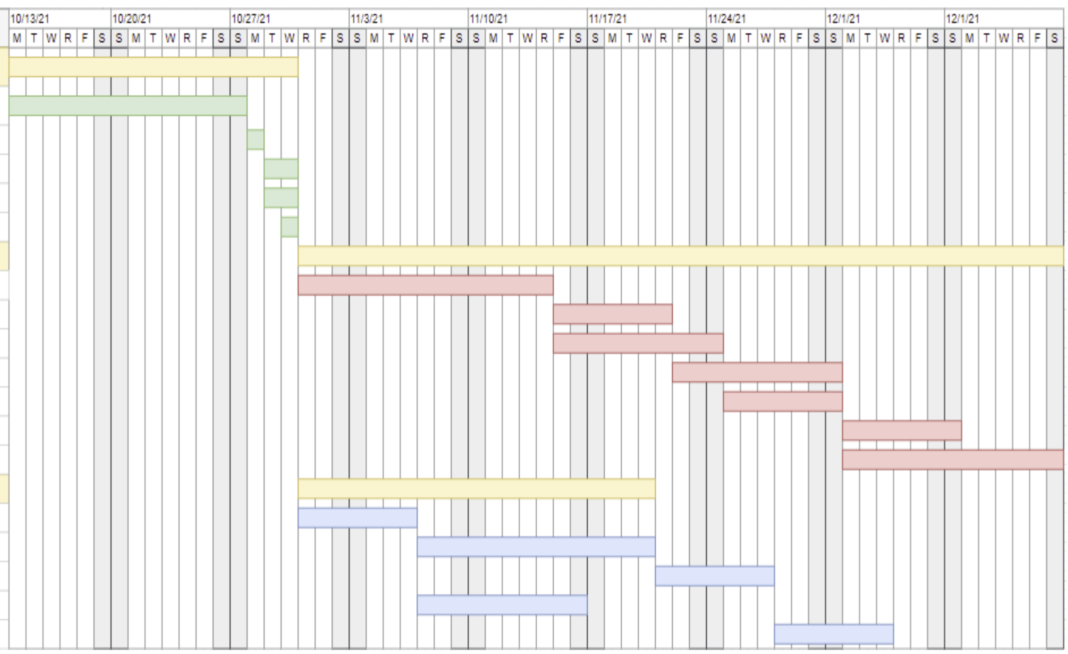
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

- Successful communication between frontend and backend
- A working use of QGIS, Leaflet.js, and other technologies
- Load watershed data that doesn't change.
- Make project available on git and vm accessible by all team members
- Provide a detailed README for the project so that all team members know how everything works together.
- Prediction with collected precipitation data without automatic updates
- Prediction with updates as data is collected by client
- 5 second page response time

Tasks will be measured by client satisfaction with these specific predefined tasks.

3.4 Project Timeline/Schedule

#	Task Name	Duration	Start	End
1	Decisions for GIS (Geographical Information Systems) software	20 days		
1.1	Select API and/or WMS (Web Mapping Service)	14 days		
1.2	Demo for client and gain approval	3 days		
1.3	Create sprints on trello	1 days		
1.4	Setup git repository for project	1 day		
1.5	Assign tasks for team members on trello	1 day		
2	Front-end Development			
2.1	Basic Front-end Framework	14 days		
2.2	Map API integration	7 days		
2.3	Color Coded Timelines	10 days		
2.4	Map Overlays	10 days		
2.5	Tabs/sections for graphs and overlays	5 days		
2.6	Precipitation graphing	7 days		
2.7	Graphing data in correct locations	14 days		
3	Back-end Development			
3.1	Create Server and database framework	7 days		
3.2	Model with static data from client	14 days		
3.3	Model with live data from client	7 days		
3.4	Well-defined endpoints	10 days		
3.5	Mapping front-end to back-end	7 days		



[1]

[1] Gnatt chart schedule of project

3.5 Risks And Risk Management/Mitigation

- Frontend development
  - Maps API integration - 0.5, we've never done maps API integration before.
    - Different map overlays - 0.2
  - UI features
    - Color coded timeline - 0.1
    - Color coded map - 0.5, we would have to find another way to integrate color coding into a map. If it isn't something we can do on QGIS, we will have to figure out a way to do so with Leaflet.js which will be significantly harder and require more knowledge of how the API works.
    - Tabs/sections for graphs and overlays - 0.05
  - Data visualization
    - Precipitation graphing - 0.2
    - Graphing data in correct locations - 0.2

- Backend development
  - Acquire data
    - Model with static data - 0.1
    - Model with live data - 0.4
  - Storing data
    - Connect server to database - 0.3
  - Developing communication model with the frontend
    - Well-defined endpoints - 0.1
    - Mapping frontend to backend - 0.3

3.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in the total number of person-hours required to perform the task.

Maps API integration	**20 hours**
Different map overlays	4 hours
UI features	**20 hours**
Color coded timeline	3 hours
Color coded map	4 hours
Tabs/sections for graphs and overlays	2 hours
Data visualization	**20 hours**
Precipitation graphing	10 hours
Graphing data in correct locations	10 hours
Acquire data	**15 hours**
Model with static data	7.5 hours
Model with live data	7.5 hours
Storing data	**8 hours**
Connect server to database	8 hours
Developing communication model with the frontend	**15 hours**
Well-defined endpoints	5 hours
Mapping frontend to backend	10 hours

[2]

[2] Number of person hours required per task

3.7 Other Resource Requirements

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

We may need to help our client find meteorological data sources that are open and free to use for the final production system. All other resources are covered by our project plan.

4 Design

4.1 Design Context

4.1.1 Broader Context

Area	Description	Examples
Public health, safety, and welfare	Our project is able to forecast the flood, and report daily precipitation, both the direct and indirect user can use it to check if their area will be affected by the flood in order to avoid any loss.	The flood forecasting tool can work as an alarm for people who live in flood areas, and people can have enough time to do prep before the flood. This tool can help with public safety and decrease the loss.
Global, cultural, and social	At the moment, we are probably focusing on flood modelling within the USA. There are a number of communities within the USA, but the central value that will be reflected by our project is safety awareness. The average citizen within the USA wants to be safe. Our flood modeling software should accurately inform users of flood hazards before they occur.	Weather alerts already exist to help protect people from natural disasters. Our model follows the same principle of safety assurance. People should get the same general message from our model as they would a flood watch or warning.
Environmental	Given our tool is weather related, it is very likely to have environmental impacts. Our hope is that it is positive, bringing more accurate flood forecasts to help better prepare for hazardous flood conditions. Perhaps the data could be wrong or misinterpreted, but this is the only negative environmental impact our team can foresee and is inherent to any tool of its kind.	Allowing more lead time to prepare and move life and property away from worst conditions. Help better understand how various rivers and watersheds respond to different weather conditions.
Economic	This will impact the economy by helping protect against loss of product value that occurs from a flood.	Since our tool helps predict flooding given rainfall and soil conditions in a certain area, people can prepare for floods by purchasing precautions before a likely flood, which would increase business to vendors that sell those precautions and help against loss of personal items of value.

[3]

[3] Our project's impact on communities / society as a whole

4.1.2 User Needs

1. Prediction users (those who are interested in artificial intelligence predictions). These users are more interested in the technical aspects of prediction using data. Useful information shown from graphs and trends would give insights that people just interested in the prediction results would not be interested.

2. Meteorological users (people who use weather prediction for forecasting). These users would be the most common, as they are interested in the actual predictions and visual aspects of the project. This group probably wants to know the odds of weather data or flooding.

4.1.3 Prior Work/Solutions

Precipitation and river flood forecasting is nothing new. Our team's main focus so far has been researching various methods and tools to implement our solution best. What makes our project unique is the inclusion of a machine learning model to make the prediction. Based on our research, AI based approaches to weather forecasting are quite rare. Most existing solutions use physics based approaches to extrapolate current conditions.

Pros
- Machine Learning approach, a good differentiation between our project and other weather prediction product available
- Nice UI, while there does exist other small projects similar to ours the UI is fairly limited

Cons
- Limited in scope

4.1.4 Technical Complexity

1. The design consists of multiple components/subsystems that each utilize distinct scientific, mathematical, or engineering principles –AND–
2. The problem scope contains multiple challenging requirements that match or exceed current solutions or industry standards.

4.2 Design Exploration

4.2.1 Design Decisions

1. We decided to use open sources instead of building everything from scratch since it will save us more time in order to focus on UI design and other features.

2. We decided to have waterfall planning with an agile development.

3. We decided to have both static and live data models.

4. We still need to make decisions on datasets and methods of mapping data to the map.

5. We will need to make decisions on testing.

4.2.2 Ideation

We used problem ideation by identifying the functional requirements of the system, then looked at similar softwares that had solutions to those problems. An example design decision would be how to display water data with some type of map. We talked about initial ideas like using a google maps API then sought out existing solutions that used it. During this time, we identified different solutions that had aspects we liked and didn't like. Through this, we found different solutions that we didn't know existed or were viable. For this we found open sourced mapping software called QGIS that specialized in this problem.

4.2.3 Decision-Making and Trade-Off

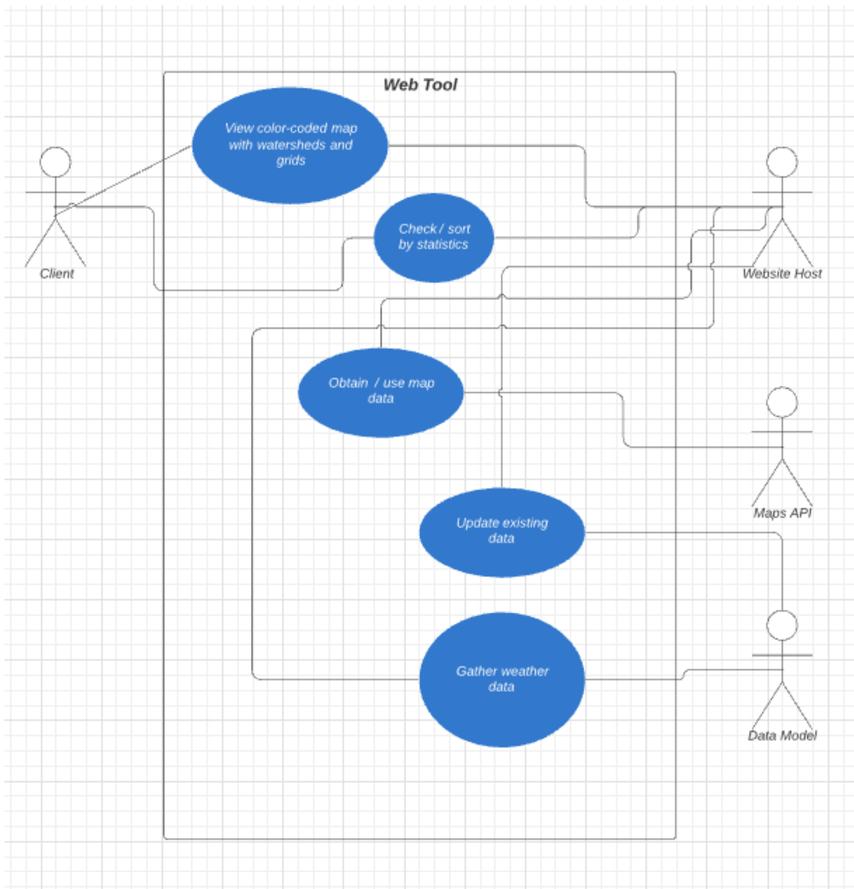
Our options for mapping the watershed were all mapping API's. We looked at the Google Maps API, but we found that the price of using the API was far too great. We eventually settled for QGIS because it is open-source and provides a way to export data to a db using PostgreSQL and POSTGIS. This appears to be the standard being used for GIS maps on the web. We then found a way to retrieve the data from the database using PHP and Leaflet.js. From there we decided to use a framework called cakePHP which is compatible with PostgreSQL. The decision to use a framework is because we want to create a professional, secure, and reliable website.

4.3 Proposed Design

We have looked into various Maps API and existing software tools with similar uses. The main thing that we have been looking for is how to integrate all the data that we have into a UI that we can customize. The main Maps API that we are looking into is QGIS. This software allows us to utilize existing GIS data and manipulate it how we see fit. We implemented a demo version of this in order to test that we can use it with other software and that we can use it for our needs. What we have found is that we can use this software along with other API's in order to fulfill the client's needs.

4.3.1 Design Visual and Description

Here's the use case diagram for our project. It describes a web based tool that displays a color coded gridded map with watersheds.

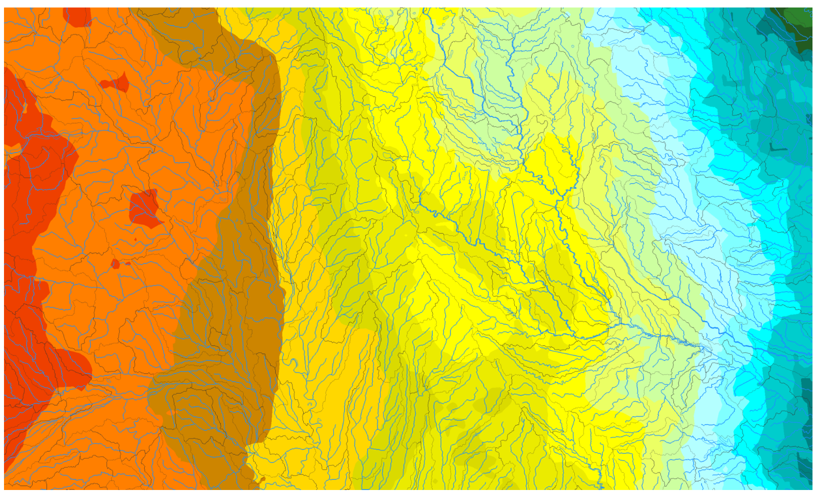


[4]

---

[4] Use case diagram

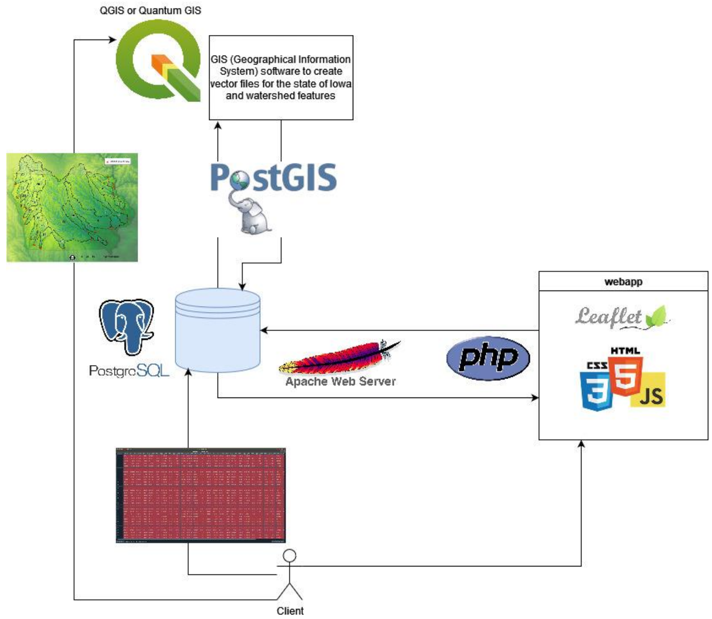
We would like to overlay different data sources such as precipitation, rivers, and watershed boundaries on a single map. This image is a demo example of gridded precipitation data with overlaid rivers. This would be shown on an interactive map with the ability to customize what is shown and navigate to various points in time.



[5]

[5] Visual aid for application use case

4.3.2 Functionality



[6]

Our project is intended to be a web application utilizing GIS software to create a visual representation of our client's AI for watersheds in the state of Iowa. In the diagram above, it shows what technologies could be used to accomplish this with open-source software. One major challenge in our project is to be able to modify data in our SQL database corresponding to our client's data.

This design satisfies the client's functional requirements of having an interactive map with selectable watersheds because the GIS software we use has the capability to create such maps, as we have seen in examples we showed to our client.

The Display of water discharge over time and color coordinating of weather data and timeline requirements are more difficult to meet with our current design. No one in our group has experience with creating GIS maps using QGIS software or something similar.

Our design satisfies all of our non-functional requirements well because our rendering of the map using Leaflet.js on the web app has great performance and is reliable. Our design is also easy to update and maintain by the client because she can connect to our database and load her data and our software will use the already existing GIS map and simply update the weather data such as precipitation. Our design is also scalable in that it can be deployed on Iowa State's domain. Once we have shown the process to create a GIS map on QGIS and deploy it to a web app, other software developers can use the same method to deploy their own maps embedded in web applications. Modifying the map itself and not just the data in the map may be more difficult for our client. It would involve learning how to use QGIS or whichever software used to create the map and could

[6] Visual aid showing how different frameworks/applications interact

involve modifying the tables within the database, which can potentially require recreating the table entirely.

4.3.3 Areas of Concern and Development

- One of our concerns has to do with how we are acquiring data. We need to be sure of where our watershed data is coming from to solidify our final design. Clarifying how the data will be obtained with our client is a high priority for resolving this concern, as we could obtain data from many sources.

- Whatever libraries / frameworks we use need to be able to display a color-coded map. Visualizing data will be a lot easier if we can use different colors to represent different things on the map. The solution to the above concern will primarily be adequate research on the libraries / frameworks we decide on.

- Another smaller concern could also be the user base we are developing for. Who is the primary audience for this design? The primary audience will have an impact on the features we include. We will need to clarify this with the client to verify who will be the primary user for the app.

- Another concern we are currently working on is figuring out what open source software to base our application on. Upon deciding that we were going to use open-source software, we started looking into what was available. We have a candidate right now, but we should definitely address this concern by doing more research on the open source software available.

- One last area of concern has to do with what kind of data we want to display. Are we only displaying flood and watershed data? Should we aim to incorporate more data into our model? How much data does the user really need? We need to clarify this with our client when we begin to actually develop the model.

4.4 Technology Considerations

- One big strength of our technology is it is all free. There will be no big financial part of our project in order to maintain it
- Another strength is that we should be able to easily map watershed data to geographical locations. As long as we have regions pertaining to our watershed data, this problem should be solved
- One trade-off of QGIS is it will probably not have as many features as a paid framework. It is still a powerful tool, but we may find a few shortcomings when we start to implement everything
    - If we absolutely need to, we can re-evaluate our design and modify it to tackle any shortcomings
- The data that our client gives us may not be “exactly” in the format we want it to be. That may end up being a weakness of our model
    - This would require us to develop an easy way to convert said data

- Another potential weakness of our project is that it needs to use a shapefile to display boundaries
    - If our client wants to use a different method to display boundaries, it could drastically change how our models interact. In the near future this will probably not happen, but in the far future it is a concern

4.5 Design Analysis

We did not adhere to the plan perfectly, but did get a simple demo working. Things we did get working:

- Basic front-end framework
    - Create watershed boundaries
    - Display simple data related to watershed (name, id, etc)
- Basic back-end framework
    - Store static data from client
    - Backend data seems to be pertaining to correct locations

Things we did not get working:

- Model live data
- Precipitation graphing
    - Analysis of watershed data in general
- Color-coded timelines
    - No extensive UI development
- Any sort of overlay

Overall, we think that our proposed design from 3.3 worked. Our main goal for our semester was to get a basic demo working, which was achieved.

To iterate over our design, we will need to assign roles to our project. We are at the point where we need to start assigning well-defined tasks to make progress. We can still use the same set of requirements from 3.3 to do this. Given that we already have a basic implementation, tackling different parts of the project should be significantly easier. Organizing our efforts will be the bigger challenge.

4.6 Design Plan

Everything in our project is building on a previous task. Because we have a solid starting point, we need to analyze our requirements and see which ones need to be completed first:

- Watersheds are selectable
    - Mouse-over coordinates, drag and click, etc
    - cakePHP/PostGIS
- For the UI tasks, we will need to interface with the selectable watersheds (with Leaflet)
- Data now needs to be graphed
    - Precipitation
    - Land cover
    - Soil properties
    - HTML/CSS/JS development

- Statistics over time can be implemented at the same time
    - Water discharge
    - Weather data
    - HTML/CSS/JS development
- We can work on making regions / statistics color coded
    - Data should change highlight color of boundaries
    - Timelines should also be color coded to aid users in understanding the application
    - HTML/CSS/JS to create UI
- Backend development will be perpetual
    - Whatever is being worked on for frontend, will need to be verified on backend
    - PostGIS/PostgreSQL

This design plan mainly involves frontend milestones. Backend development should be working on the same "task" as frontend (graphing precipitation, weather data over time, etc) to ensure manual testing can occur. As highlighted in our requirements, it is important that we organize our work well to avoid deadlock (something mandatory is not completed yet). This outline emphasizes the functional requirements we need to complete; the model in 3.3 is a more comprehensive idea of our constraints.

5 Testing

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, power system, or software.

The unique challenge we face as a team is coming up with the best method for testing how well our GIS map satisfies the functional requirements and nonfunctional requirements that our client has given us. For example, testing how interactive our map is and whether it satisfies the client's expectations requires qualitative assessments of the features available with our application.

5.1 Unit Testing

Our QGIS testing will be focused on getting individual features to work. We want each feature to be a map layer, so each individual feature would be a unit. For example, watersheds and grids on the map, precipitation and weather data overlaid, etc...

Testing database API calls. We want to put our data in a database and make sure we can access it through the frontend web app and in QGIS. We would have to test that we can call individual pieces of data, as well as test that other data isn't corrupted because of that.

Our web application will have unit testing through the frontend. Making sure all UI features work as intended with test values.

We would use IDE testing tools with each of these. Our tech stack includes JavaScript, HTML/CSS on the frontend, Python with QGIS, and PHP in the backend. Since most of our unit tests would be run through the frontend, we would use VS Code or whatever IDE we pick to run unit tests using their specified frameworks/libraries. We would also use Postman to test on a mock server and test backend.

5.2 Interface Testing

One interface we will have to use is a maps interface. We have defined what our map interface will need to be, and we should now have ways we can test it. Testing the interface will probably consist more of testing communication between components. Depending on which map implementation we decide upon, we might end up having specific test cases be different:

- Verifying that functions within the interface are being called correctly (mocking)
- Checking that data retrieved is being processed efficiently (responsiveness of application)
- Verifying integrity of data actually being displayed (interface is actually doing something with data)
- Testing that the map view can be effectively navigated (seeing that our maps framework is reliable)

Another interface we will have to implement and test is how our data will be obtained. We will need a solid model for storing and obtaining data when requested. Testing our implementation of the data storage interface will primarily consist of:

- Verifying that data is being obtained and is present in a database (manually)
- Having a well-defined set of APIs (test endpoints thoroughly)
- Testing actual communication between frontend and backend (verify data received from frontend API calls)
- Verify that backend controllers and services are communicating with database (mocking)

5.3 Integration Testing

Our two critical integration paths for our design is integrating QGIS into a web application for the front end and integrating the database into a usable format for the back end. The first one is critical because the client has a requirement for the data to be in a visual format that is visible in a web application. The criticality for the second path comes from that if we can't get the right data from the database, the map will no longer have correct or accurate data, which fails one of our requirements. We will test the QGIS by comparing what we know should be on what location and seeing whether the data is correct. For testing the database integration, we will make sure that all the data is parsed correctly and no decrease in accuracy occurs. Most likely, other tools will not be used.

5.4 System Testing

Our testing strategy for the entire system will be to go through the functional and non-functional requirements and check to see how well they are satisfied. We would be testing how the multiple integrations work in conjunction. We would use a unit/interface test that coordinates with a specific UI feature. Since our UI comprises an integrated maps component into our frontend, it would also be testing that integration.

5.5 Regression Testing

We will have to use separate unit tests for each task, continuous integration will be involved to make sure they do not break the old functionality, and keeping track of the old and new additions is necessary. We need to ensure implemented critical features for both frontend and backend will not break. It will be driven by project functional, nonfunctional requirements and testing tools.

5.6 Acceptance Testing

Given many of our functional requirements are UI based, it will be very difficult to have automated acceptance testing. Our plan is to instead manually evaluate the criteria with our client.

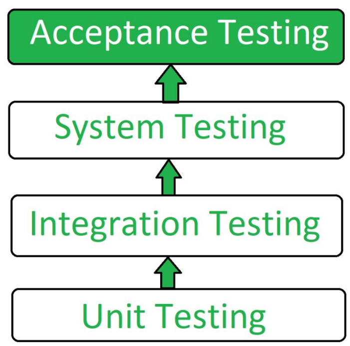
5.7 Security Testing (if applicable)

Security testing may involve scanning our web server internally for any XSS or SQL injection that may take place. Also need to ensure that the integrity of our client's data is kept. The data points for precipitation, land cover, watershed boundaries, etc. are okay to be accessed, but any of the machine learning algorithms that generate the data need to be kept confidential.

5.8 Results

One result of our testing is it allows us to make changes while ensuring previous functionality does not break. Another result is to help us to establish clear expectations from the start and use testing as a verification.

This image shows the various levels of abstraction within our testing plan. It goes from unit testing, which is very selective individual component level testing, up to acceptance testing, which tests high level functional requirements.



[7]

[7] The stages of testing for this project

6 Implementation

Any preliminary implementation plans would stem from needing to allocate tasks. Given that we have a basic implementation of our project already, we will need to tackle the more challenging parts of our design plan we could not get to this semester. Work will likely be divided into frontend and backend portions.

At this point those working on the project should have some experience with the frameworks we are using. If needed they can look at more tutorials / study the source code to figure out how the project will be laid out. It will be important to have a good understanding of the project before it is worked on.

Actual implementation will follow a lot of what was outlined in the design. Implementations will always be building on previous tasks. Most of the preliminary implementation (basic demo of app) we have done already will make the actual implementation a lot smoother.

7 Professionalism

This discussion is with respect to the paper titled " Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment", *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

7.1 Areas of Responsibility

Pick one of IEEE, ACM, or SE code of ethics. Add a column to Table 1 from the paper corresponding to the society-specific code of ethics selected above. State how it addresses each of the areas of seven professional responsibilities in the table. Briefly describe each entry added to the table in your own words. How does the IEEE, ACM, or SE code of ethics differ from the NSPE version for each area?

Area	Description	Differences
Work competence	Perform work of high quality, integrity, timeliness, and professional competence.	NSPE puts an emphasis on avoiding deceptive acts and being honest while performing within a competence. IEEE standards allow you to still work on areas where qualifications are limited but only after full disclosure of pertinent limitations;
Financial responsibility	Deliver products and services of realizable value and at reasonable costs.	NSPE discusses being truthful and acting in good faith. IEEE puts emphasis on value and cost of delivering products and making sure the client is not being cheated
Communication honesty	Report work truthfully, without deception, and understandable to stakeholders.	NSPE has the same description.
Health, safety, and well-being	Minimize risks to safety, health, and well-being of stakeholders.	Minimize risks while keeping safety in mind. The well-being of engineers and stakeholders should both be valued.
Property ownership	Respect property, ideas, and information of clients and others.	NSPE has the same description

Sustainability	Protect the environment and natural resources locally and globally.	Use sustainable development to keep future generations in mind. IEEE discusses the use of natural resources and protecting the environment.
Social responsibility	Produce products and services that benefit society and communities.	Honor and reputation are also a part of social responsibility, according to the NSPE. IEEE discusses making sure the product has a benefit to society and is not a detriment.

[8]

7.2 Project Specific Professional Responsibility Areas

For each of the professional responsibility area in Table 1, discuss whether it applies in your project's professional context. Why yes or why not? How well is your team performing (High, Medium, Low, N/A) in each of the seven areas of professional responsibility, again in the context of your project. Justify.

Work Competence: Each member should be working in the areas that they are the most competent. On top of that, each member should be transparent in their activities (not doing anything to sabotage or set back the project). We are performing fairly well in this area, and should keep playing to each other's strengths and weaknesses.

Financial Responsibility: Currently, this area doesn't really apply to our project. We are in the very early stages of the project, so we are not developing for a large user base. Our primary goal is to get things the way our client wants them. We do not have to explicitly worry about expense vs. profit yet. We are performing pretty low in this category, but are making up for it in other areas.

Communication Honesty: Our stakeholders include a client that needs to be informed of where we are in the project, as well as a professor who has tasked the client with their project. So, this is another important area of professionalism for us. Our tasks should be communicated clearly, and we should not lie to our clients that are invested in making our project a reality. We are performing high in this category, we have weekly contact with our client.

Health, Safety, and Well-Being: Because our project is a software engineering solution, we really don't have much safety-wise to be concerned about. We may be concerned about the general security of our application, but at this point in time it makes more sense to allocate more resources to other professional areas. We are performing somewhat low in this area too, but are considering health and safety where we have to.

[8] Professional areas descriptions and differences

Property Ownership: The work we are doing is truly owned by our clients. So, this is another important area of professionalism for our project. We can be credited for creating the project, but in the end, the work we are doing will not be owned by us. We need to keep an emphasis on crediting the open-source software that we use. It will likely be passed on to other groups to improve our model. We are performing highly in this area as we constantly check with the client to ensure the project looks the way it should.

Sustainability: While our work does have to do with tracking environmental data for prediction, our project doesn't focus on employing tactics to help the environment nor does it use many resources that may impact the environment. In this case the team is performing fairly low, but given the context of our project our attention is needed elsewhere.

Social responsibility: We will need to make sure our project has benefits for society and community, forecast danger and decrease the loss.

7.3 Most Applicable Professional Responsibility Area

Our team has demonstrated high proficiency in using properly licensed APIs, technology, and data as well as ensuring the best security practices are followed. These responsibilities are areas where we could potentially cut corners to save time, likely without the client knowing. It is through our professional integrity that we make sure these areas are properly dealt with and communicated honestly to our client.

8 Closing Material

8.1 Discussion

- The basic requirements of our project have been met. We have created a basic app with selectable watersheds
  - Frontend and backend portions of app are functional
  - Leaflet, cakePHP
  - PostGIS, PostgreSQL
- The more advanced portions of our project have not yet been met
  - Graphing data
  - Weather data over time
  - Water discharge
  - Whatever other data we think is meaningful

The parts of our project that we needed in our prototype are functional. So, the parts of our project that prove our design is valid are finished.

8.2 Conclusion

The work we have done thus far was focused on getting a prototype working. A lot of design and research went into our project this semester. In turn, near the end of the semester, we started to come up with a basic demo of our app. Our main goals were to:

- Do basic setup of project
  - Leaflet, cakePHP
  - PostGIS, PostgreSQL
- Make watersheds selectable
- Display watershed data
- Graph watershed data
- Graph weather data / statistics over time

To achieve the goals that we did not meet this semester, we will continue to follow the plan in 3.3. The main constraints to achieving these goals were unfamiliarity with the frameworks we were using. Now that we have solidified our model (and know it is a true solution), we are much more prepared to tackle the rest of the problems introduced by our project.

8.3 References

Any technical references were already cited in the professionalism section. We are not claiming any of the software / frameworks to be ours

8.4 Appendices

All important documents are already present throughout the design document. See figures page for more information on included diagrams / tables

8.4.1 Team Contract

Team Members:

1) Ani Manjunath
2) Siyu Wang
3) Quinn Conrad
4) Kylus Pettit-Ehler
5) Ryan Thompson
6) Gabriel Rau
7) Eric Korneisel

Team Procedures

Day, time, and location (face-to-face or virtual) for regular team meetings:

Face-to-face meetings every Wednesday at 2:00. Meetings with client weekly/bi-weekly as desired on Mondays at 2:15-3:00.

2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):

GroupMe, email, and face-to-face meetings.

3. Decision-making policy (e.g., consensus, majority vote):

Consensus.

4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

Google Doc in our shared drive.

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:

Everyone should show up to all meetings unless communicated otherwise.

2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:

Completing tasks by deadlines unless unexpected delays, which have to be communicated to the rest of the team.

3. Expected level of communication with other team members:

Keeping up with GroupMe and emails. Not expected to reply unless there is a question or concern directed towards a specific team member. They have a 24 hour buffer to reply.

4. Expected level of commitment to team decisions and tasks:

All members are expected to participate in team discussions and decision-making.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction,

individual component design, testing, etc.):

2. Strategies for supporting and guiding the work of all team members:

Reminders via GroupMe/email

Meetings - weekly or as desired

Coordination with other members or TA when necessary

3. Strategies for recognizing the contributions of all team members:

Keeping track of role assignments.

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the

team.

Ani: Frontend development/full-stack development. Java, C# .NET, C, AngularJS, HTML/CSS, Verilog, VHDL, SQL

Kylus: Frontend / Backend development. Java, Typescript, C, Spring, Angular, HTML / SCSS, Verilog, VHDL.

Ryan: Backend/full-stack development. Java, AngularJS, C, HTML, Python, SQL, Spring, Javascript, VHDL.

Siyu: Backend development. Java, C, SQL , HTML/Javascript/CSS, Spring,Verilog

Quinn: Full-stack development, Java, Javascript, C, C++, Python, HTML/CSS

Gabe: Backend development. Java, C, C++, C#, MySQL

Eric: Frontend/Backend development. Java, C, C++, HTML, SQL, Javascript

2. Strategies for encouraging and support contributions and ideas from all team members:

Actively participating in team decision-making

Offering extra help when necessary

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will

a team member inform the team that the team environment is obstructing their

opportunity or ability to contribute?)

We have team members assigned to those roles. We will discuss during meetings or via GroupMe.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

Get a functional design of the product and begin implementation.

2. Strategies for planning and assigning individual and team work:

First-come first-serve.

3. Strategies for keeping on task:

Consequences for Not Adhering to Team Contract

Active deadlines and constant communication/updates.

1. How will you handle infractions of any of the obligations of this team contract?

Discuss face-to-face via team meetings to be sure that any conflicts get resolved.

2. What will your team do if the infractions continue?

3 strike rule, then go to TA or professor.

****

a) I participated in formulating the standards, roles, and procedures as stated in this contract.

b) I understand that I am obligated to abide by these terms and conditions.

c) I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.

1) Anirudh Manjunath	DATE: 12/5/21
2) Siyu Wang	DATE: 12/5/21
3) Ryan Thompson	DATE: 12/5/21
4) Quinn Conrad	DATE: 12/5/21
5) Kylus Pettit-Ehler	DATE: 12/5/21
6) Gabriel Rau	DATE: 12/5/21
7) Eric Korneisel	DATE: 12/5/21